

SVM

```
In [75]: import numpy as np
import pandas as pd
import seaborn as sns
from sklearn.preprocessing import LabelEncoder
from sklearn import model_selection
from sklearn import preprocessing
% matplotlib inline
```

```
In [76]: adult_df = pd.read_csv('adult.data',
                                header = None, delimiter=' ', engine='python')
```

```
In [77]: adult_df.columns = ['age', 'workclass', 'fnlwgt', 'education', 'education_num',
                             'marital_status', 'occupation', 'relationship',
                             'race', 'sex', 'capital_gain', 'capital_loss',
                             'hours_per_week', 'native_country', 'income']
```

```
In [78]: adult_df.isnull().sum()
```

```
Out[78]: age                0
workclass              1836
fnlwgt                 1843
education              1843
education_num          1843
marital_status         1843
occupation             1843
relationship           1843
race                   1843
sex                    1843
capital_gain           1843
capital_loss           1843
hours_per_week         1843
native_country         1843
income                 1843
dtype: int64
```

```
In [79]: for value in ['workclass', 'education',
                    'marital_status', 'occupation',
                    'relationship', 'race', 'sex',
                    'native_country', 'income']:
    print (value, ":", sum(adult_df[value] == '?'))
```

```
workclass : 1836
education : 0
marital_status : 0
occupation : 1843
relationship : 0
race : 0
sex : 0
native_country : 583
income : 0
```

```
In [80]: adult_df_rev = adult_df
for value in ['workclass', 'education',
             'marital_status', 'occupation',
             'relationship', 'race', 'sex',
             'native_country', 'income']:
    adult_df_rev[value].replace(['?'], [adult_df_rev.describe(include='all')[value][2]],
                               inplace=True)
```

```
In [81]: adult_df_rev.sample(3)
```

```
Out[81]:
```

	age	workclass	fnlwgt	education	education_num	marital_status	occupation	relationship	race	sex	capital_gain	ca
21661	21	Private	124589	HS-grad	9	Never-married	Sales	Own-child	White	Female	0	0
5465	49	Self-emp-inc	30751	Assoc-voc	11	Divorced	Exec-managerial	Not-in-family	White	Male	0	0
25219	74	Private	192290	10th	6	Married-civ-spouse	Adm-clerical	Husband	White	Male	0	0

```
In [82]: le = preprocessing.LabelEncoder()
workclass_cat = le.fit_transform(adult_df.workclass)
education_cat = le.fit_transform(adult_df.education)
marital_cat = le.fit_transform(adult_df.marital_status)
occupation_cat = le.fit_transform(adult_df.occupation)
relationship_cat = le.fit_transform(adult_df.relationship)
race_cat = le.fit_transform(adult_df.race)
sex_cat = le.fit_transform(adult_df.sex)
native_country_cat = le.fit_transform(adult_df.native_country)
```

```
In [83]: adult_df_rev['workclass_cat'] = workclass_cat
adult_df_rev['education_cat'] = education_cat
adult_df_rev['marital_cat'] = marital_cat
adult_df_rev['occupation_cat'] = occupation_cat
adult_df_rev['relationship_cat'] = relationship_cat
adult_df_rev['race_cat'] = race_cat
adult_df_rev['sex_cat'] = sex_cat
adult_df_rev['native_country_cat'] = native_country_cat
```

```
In [84]: dummy_fields = ['workclass', 'education', 'marital_status',
                        'occupation', 'relationship', 'race',
                        'sex', 'native_country']
adult_df_rev = adult_df_rev.drop(dummy_fields, axis = 1)
```

```
In [85]: adult_df_rev.sample()
```

```
Out[85]:
```

	age	fnlwgt	education_num	capital_gain	capital_loss	hours_per_week	income	workclass_cat	education_cat	marital_ca
30207	30	487330	10	0	0	30	<=50K	3	15	2

```
In [86]: adult_df_rev = adult_df_rev.reindex_axis(['age', 'workclass_cat', 'fnlwgt', 'education_cat',
                                                  'education_num', 'marital_cat', 'occupation_cat',
                                                  'relationship_cat', 'race_cat', 'sex_cat', 'capital_gain',
                                                  'capital_loss', 'hours_per_week', 'native_country_cat',
                                                  'income'], axis= 1)

adult_df_rev.sample()
```

```
Out[86]:
```

	age	workclass_cat	fnlwgt	education_cat	education_num	marital_cat	occupation_cat	relationship_cat	race_cat	sex_cat
9248	38	1	745768	15	10	0	3	4	2	0

```
In [87]: num_features = ['age', 'workclass_cat', 'fnlwgt', 'education_cat', 'education_num',
                        'marital_cat', 'occupation_cat', 'relationship_cat', 'race_cat',
                        'sex_cat', 'capital_gain', 'capital_loss', 'hours_per_week',
                        'native_country_cat']
```

```
In [88]: features = adult_df_rev.values[:, :14]
target = adult_df_rev.values[:, 14]
features_train, features_test, target_train, target_test = train_test_split(features, target, test_size = 0.40, random_
state = 10)
```

```
In [89]: clf = svm.SVC()  
         clf.fit(features_train, target_train)  
         target_pred = clf.predict(features_test)
```

```
In [90]: print(accuracy_score(target_test, target_pred, normalize = True)*100)  
76.0383877159
```

```
In [ ]:
```